# CAST: Efficient Traffic Scenario Inpainting in Cellular Vehicle-to-Everything Systems

Liang Zhao, Member, IEEE, Chaojin Mao, Shaohua Wan, Ammar Hawbani, Ahmed Y. Al-Dubai, Senior Member, IEEE, Geyong Min, and Albert Y. Zomaya, Fellow, IEEE

Abstract—As a promising vehicular communication technology, Cellular Vehicle-to-Everything (C-V2X) is expected to ensure the safety and convenience of Intelligent Transportation Systems (ITS) by providing global road information. However, it is difficult to obtain global road information in practical scenarios since there will still be many vehicles on the road without onboard units (OBUs) in the near future. Specifically, although C-V2X vehicles have sensors that can perceive their surroundings and broadcast their perceived information to the C-V2X system, their line-of-sight (LoS) is limited and obscured by the environment, such as other vehicles and terrain. Besides, vehicles without OBUs cannot share their perceived information. These two problems cause extensive areas with unperceived information in the C-V2X system, and whether vehicles are in these areas is unknown. Thus, extending the perceivable range of the limited scenario for C-V2X applications that require global road information is necessary. To this end, this paper pioneers investigating the scenario inpainting task problem in C-V2X. To solve this challenging problem, we propose an effiCient trAffic Scenario inpainTing (CAST) solution consisting of a generative architecture and knowledge distillation, simultaneously considering the inpainting precision and computation efficiency. Extensive experiments have been conducted to demonstrate the effectiveness of CAST in terms of Precise Inpaint Rate (PIR), Rough Inpaint Rate (RIR), Lane-Level Inpaint Rate (LLIR), and Inpaint Confidence Error (ICE), paving the way for novel solutions for the inpainting problem in more complex road scenarios.

Index Terms—C-V2X Scenario Inpainting, Generative Architecture, Knowledge Distillation, Intelligent Transportation Systems.

## 1 Introduction

With the development of Intelligent Transportation Systems (ITS), Cellular Vehicle-to-Everything (C-V2X) has been garnering significant research interests due to its potential to support ubiquitous communication services for ITS [1, 2, 3]. C-V2X enables vehicles to communicate with other devices in ITS by equipping the on-board unit (OBU), which ensures vehicle safety and improves traffic efficiency, as well as providing recreation applications [4, 5, 6]. Specifically, ITS applications will greatly benefit from C-V2X by integrating line-of-sight (LoS) and non-line-of-sight (NLoS) range information so as to gain global road information, i.e., all vehicle states on the road.

The aforementioned visions, however, are all based on the same premise that every vehicle on the road is equipped with an OBU that works properly, allowing vehicles to broadcast their states for building global road information for the C-V2X system. Nevertheless, vehicles without OBUs will exist on the road for a long time due to the slow penetration rate of C-V2X vehicles. The coexistence of vehicles equipped with/without OBUs on the road makes it unrealistic to acquire global road information, namely, the Missing Information problem in C-V2X. Specifically, the information about vehicles without OBUs and not appearing in the LoS of vehicles with OBUs is unobservable, i.e., missed information in C-V2X, for the C-V2X system since these vehicles cannot broadcast their information to others. Correspondingly, the information about parts of the environment in the surroundings of these vehicles is also unobservable. The missing information in these unperceived areas (UPAs) will lead to the unexpected invalidation of a huge number of C-V2X applications. C-V2X services, in particular, that need global road information will be unavailable when the vehicle is unable to obtain comprehensive information about its surroundings [7]. For instance, in the classic forward collision warning scenario [8], the vehicle equipped with OBU also cannot perceive an obscured vehicle ahead if neither vehicle in front is equipped with a C-V2X device. Thus, the forward collision warning scenario would no longer alarm the danger at the time in this situation. Moreover, when it comes to efficient cooperative lane changes for C-V2X vehicles [9], the missing information in UPAs would impede the C-V2X system from improving traffic congestion. Therefore, it is crucial to eliminate the negative impact of the missing information in UPAs to improve the reliability and efficiency of the C-V2X system.

Unfortunately, the missing information problem in UPAs of C-V2X scenarios is still unsolved in the existing works. Thus, we place a great deal of emphasis on this problem in this paper and we call it the C-V2X scenario inpainting task

Please note that this work includes a supplementary file.

Liang Zhao, Chaojin Mao, and Ammar Hawbani are with the School of Computer Science, Shenyang Aerospace University, Shenyang, China. Liang Zhao is also with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China. (e-mail: lzhao@sau.edu.cn, maochaojin@stu.sau.edu.cn, anmande@ustc.edu.cn)
Shaohua Wan is with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China. (e-mail: shaohua.wan@ieee.org)
Ahmed Y.Al-Dubai is with the School of Computing, Edinburgh Napier University, UK. (e-mail: a.al-dubai@napier.ac.uk)
Geyong Min is with the Department of Computer Science, University of Exeter, UK. (e-mail: G.Min@exeter.ac.uk)
Albert Y. Zomaya is with the School of Computer Science, University of Sydney, Australia. (e-mail: albert.zomaya@sydney.edu.au)
Ammar Hawbani and Chaojin Mao are the corresponding authors.

that aims to recover the missing information, especially the position of vehicles located in UPAs, focused in this paper, for C-V2X scenarios. Through analysis, there are two feasible ways to settle this task, (1) deploying sufficient monitors on the roadside to obtain timely road information that cannot be collected by the sensors of vehicles with OBUs, which can effectively supplement the information obtained by the vehicle with OBU; (2) building a road information inpainting model that generates overall information on the road by analyzing the known information.

The aforementioned first method is feasible in some cases by deploying enough monitors to cover the overall area on the road. However, the high cost and easily affected sensing distance of those monitors due to the uncontrollable weather or environment limit the effectiveness of this method in most cases. Specifically, in a heavy fog day or other extreme weather, the range of sensors in both monitors and vehicles is extraordinarily limited, making it difficult to effectively perceive the information on the road. In contrast, the aforementioned second method is more effective in the majority of situations due to its economy and ability to adapt to the environment. Therefore, we focus on the second method in this paper, which builds an artificially intelligent algorithm to infer information about UPAs in C-V2X by analyzing known road information. However, it is challenging to effectively inpaint the C-V2X scenario based on incomplete information. Fortunately, the significant success of video inpainting [10] and vehicle trajectory prediction [11] tasks, inferring unknown information based on known information, inspires us to make efforts in this C-V2X scenario inpainting. Nevertheless, it is infeasible to directly utilize the existing methods from video inpainting and vehicle trajectory prediction tasks in the C-V2X scenario inpainting task due to two crucial disadvantages: 1) vehicle trajectory prediction methods do not consider the UPAs, but only focus on the paradigm that generates the future state of vehicles based on the historical state of vehicles; 2) the video inpainting methods ignore the impact of the interaction between vehicles and the characteristics of the unidirectional traffic flow on the road. Hence, it is necessary to come up with a new method to adapt the features of the C-V2X inpainting task.

As a pioneering work in C-V2X inpainting, we focus on the recovery of the position of vehicles located in UPAs because it is the most significant aim of this task. Given there are some inherent correlations between the behavior of vehicles and it is feasible to inpaint UPAs by learning these relationships, we propose a generative adversarial network and knowledge distillation-based effiCient trAffic Scenario inpainTing (CAST) model to accurately inpaint the C-V2X scenario as well as consider the computation efficiency, to restore the position of vehicles in UPAs in real-time. The main contributions of this paper are summarized as follows.

- This paper pioneers to address the Missing Information problem in C-V2X scenarios due to the coexistence of vehicles with/without OBUs and concludes this problem as a C-V2X scenario inpainting task to facilitate other researchers attending in solving this problem.
- This paper creates a local and global generative adversarial model to realize the C-V2X scenario inpainting. Given the different degrees of influence of diverse perceived areas (PAs) on diverse UPAs, a sorting algorithm is proposed for UPAs to guide the generating process of the local generative adversarial model. Due to the lag of interaction between vehicles, a group trajectory prediction (GTP) model is proposed to further extract the potential interaction between vehicles in PAs and UPAs.
- Given the limited computation capacity of edge devices deploying the inpainting model and the low latency requirement of some C-V2X applications, the inpainting efficiency is also a key factor for practical implementation. Thus, an architecture based on cross-model distillation and self-distillation is proposed to ensure inpainting precision and efficiency simultaneously.
- To measure the performance of the proposed solution for the C-V2X scenario inpainting task, this paper proposes four metrics, including Precise Inpaint Rate (PIR), Rough Inpaint Rate (RIR), Lane-Level Inpaint Rate (LLIR), and Inpaint Confidence Error (ICE), respectively.

The rest of this paper is organized as follows. Section 2 reviews the literature. The C-V2X scenario inpainting problem is addressed in Section 3. In Section 4, we detail the method of processing the dataset. In Section 5, we further introduce the architecture of CAST. In Section 6, we evaluate the performance of the CAST. Finally, Section 7 summarizes this paper.

## 2 Related Work

Considering the fact that the C-V2X scenario inpainting is a brand-new task first introduced in this paper, we discuss and summarize the two crucial fields that inspired us to work out this problem, including the video inpainting and the vehicle trajectory prediction task, in this section.

### 2.1 Video Inpainting

In the past decades, image inpainting methods have achieved great success[12, 13, 14]. However, it is difficult to directly apply the image inpainting method to the video frame inpainting due to a lack of consideration of temporal consistency. Video inpainting methods can be divided into two types, including traditional methods and deep learning-based methods. The patch-based single image inpainting method was extended to the video inpainting task at the beginning [15, 16]. However, traditional methods are confronted with the high computational cost problem. Thus, deep learning-based methods are proposed to provide more efficient and effective ways. Wang et al. [10] developed a method by merging 2D and 3D convolution to recover the unknown information of the video. Li et al. [17] designed an end-to-end framework for flow-guided video inpainting by designing an architecture with three modules. Given the effectiveness of GAN in guiding image generation, many works utilize GAN to improve the result of video inpainting. Wu et al. [18] proposed a spatial-temporal Nested GAN (STN-GAN) for video inpainting by comprehensively considering the temporal and spatial information. Zou et al. [19] designed an end-to-end deep network that contains a new temporal shift-and-aligned module (TSAM), which utilizes the GAN loss function to train the model. Ke et al. [20] augmented the process for training the temporal patch-based GAN by utilizing a novel multi-class discriminator that

Table 1
comparison of existing work

| Feature<br>Related work | Vehicle interaction | UPAs | Inferring unknown | Unidirectional flow | Global interaction | Prediction priority |
|---|---|---|---|---|---|---|
| Vehicle trajectory prediction | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Video inpainting | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ |
| CAST | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

includes a spatiotemporal attention module (STAM). Nguyen et al. [21] proposed an improved GAN-based method for image inpainting by utilizing multiple generators and discriminators. Cha et al. [22] presented a GAN-based method that uses a dynamic attention map (DAM-GAN) to reduce pixel inconsistency disorder resulting from fake textures. On the basis of SN-PatchGAN [23], Zhang et al. [24] proposed the Adaptive Style Fusion Network (ASFN) for video inpainting. This method significantly improves the model convergence and the quality of inpainting.

### 2.2 Trajectory Prediction

The existing vehicle trajectory prediction methods can be classified as physics-based methods, maneuver-based methods, and interaction-aware-based methods [11, 25, 26], respectively. The physics-based method can achieve fast vehicle trajectory prediction. However, the physics-based method can merely be used for short-time and low-precise vehicle trajectory prediction because it only takes the physical regular of the vehicle and the temporary vehicle state into account for predicting the future vehicle trajectory but overlooks the impact of the environment [27]. The maneuver-based method regards each vehicle as an independent entity that contains a series of maneuvers in each movement period. Wang et al. [28] used a Bayesian inference network to determine the driving behavior of the current vehicle and, at the same time, to judge the predicted trajectory of the vehicle based on the physical information. Mandalia et al. [29] proposed the method of using a support vector machine to judge vehicle lane change decisions and predict vehicle trajectory. Yoon et al. [30] presented a method of realizing lateral motion prediction (MLP) by using a multi-layer perceptron model. Although both the physics-based method and the maneuver-based method are efficient and interpretable, the precision of the predicted vehicle trajectory in the two methods is not up to expectations due to the neglect of the interactions between the vehicle and its surroundings. Therefore, researchers gradually turned their attention to the interaction-aware-based method, which takes full advantage of the interaction between the vehicle and its surroundings. In addition, recent works propose to utilize deep learning to solve the above complex problem thanks to the development of big data and hardware. Park et al. [31] designed an encoder and decoder-based architecture that uses a Long Short Term Memory (LSTM) encoder to analyze past trajectory information, an LSTM decoder to generate future trajectory information, and beam search to search the final result on an occupancy grid graph. Cai et al. [32] presented a model that can extract the deep relationships of surrounding vehicles.

### 2.3 Summary

In light of the aforementioned related works, we can make the following preliminary summary: (i) the major video inpainting methods utilize the known temporal and spatial information in the frame sequence of video for inpainting; (ii) the prevailing vehicle trajectory prediction methods take full advantage of the time series and the spatial interaction between the target vehicle and its surroundings to predict the future trajectory of the vehicle. As shown in Table 1, methods for both domains are all able to infer unknown information by the known information. In addition, we understand the C-V2X scenario inpainting task based on the above insights. First, the C-V2X scenario inpainting task aims to infer the position of unperceived areas on the road based on known information, which is similar to the video inpainting task. However, the interaction between the objects in the C-V2X scenario is global, which can impart one by one, and the movement of the objects in the C-V2X scenario is directed. Furthermore, the current state of the vehicles in the known areas is related to their historical state, as well as the interaction between the vehicles in the known and unperceived areas, which is similar to the vehicle trajectory prediction task. Nevertheless, the state of the vehicles in the unperceived areas is unknown. In addition, the methods in both domains do not consider the prediction priority caused by the diverse strength of interaction between perceivable areas and unperceived areas. Inspired by the aforementioned insights, we reach some new ideas to solve the above problems, (i) a GAN-based model will be proposed to extract the temporal and spatial information for inpainting the C-V2X scenario; (ii) a priority sort algorithm will be designed to enhance the effect of the state of the known areas; (iii) the group vehicle trajectory prediction model will be designed to extract the interaction between the vehicles in the perceivable and unperceived areas. Moreover, additional contributions will be made to improve the performance and effectiveness of our solution.

## 3 Problem Formulation

Given the Information Missing problem in the C-V2X system due to the limited penetration rate of OBUs, the vision promised by C-V2X cannot be ensured. In order to tackle this issue, we propose CAST for realizing the scenario inpainting of the C-V2X system by taking advantage of the known information of C-V2X, which jointly considers the inpainting accuracy and computation efficiency. In this section, we introduce the basic preliminaries of CAST. Firstly, we describe the scenario without the global insight into the road in C-V2X. Then, a comprehensive outline of CAST is given.

### 3.1 C-V2X Scenario

In this paper, we take into account that both vehicles with OBU and without OBU co-exist on the road. As shown at the

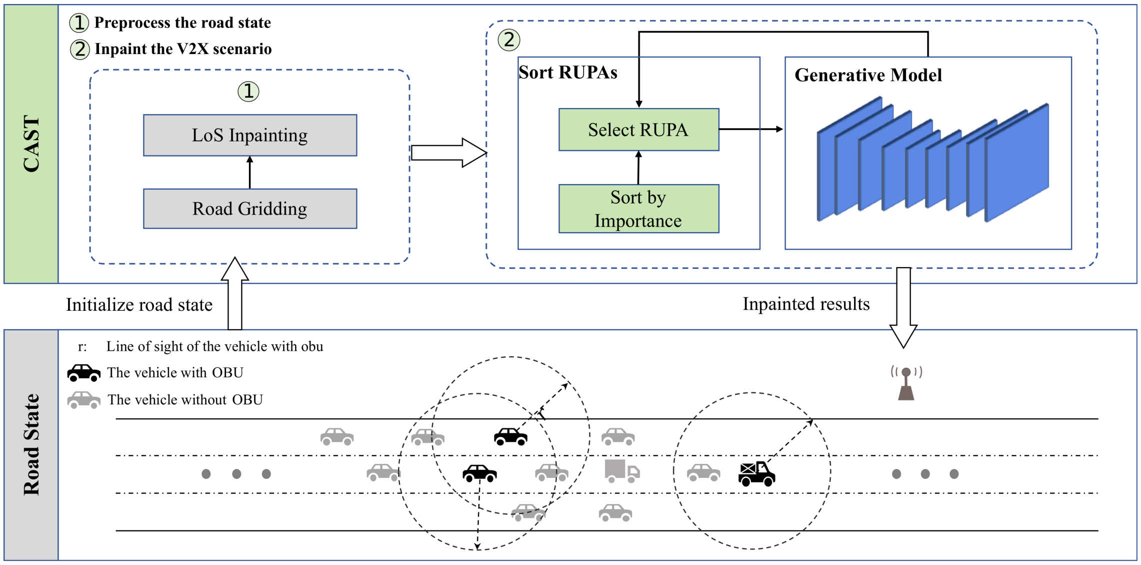


Figure 1. The architecture of CAST.

bottom of Fig. 1, black vehicles are equipped with OBUs, and gray vehicles are without OBUs. It is assumed that every vehicle with OBU can perceive the information of its surroundings, such as the state of other vehicles, by using its sensors that are limited by the environment and broadcast its own and surrounding information to the C-V2X system. In addition, we suppose that the LoS of each vehicle with OBU is $r$, which varies depending on the environment. For example, under foggy conditions, the LoS of the sensor must be shorter than that under sunny conditions. Moreover, the LoS of the vehicle can be obstructed by other vehicles or obstacles, which further limits its perceived range. Thus, there are two categories of areas on the road, unperceived area (UPA), and perceived area (PA), respectively. This paper aims to inpaint the UPAs, namely infer the position of unknown vehicles in the UPAs, to extend the range of PAs by taking advantage of the states of the vehicles in PAs.

It is obvious that the state of vehicles is affected by each other. For example, vehicles would decelerate if there is a vehicle in its front and the speed is lower than it. With the important insight above, we fully utilize the states of vehicles in PAs to extract the interaction between vehicles, so as to inpaint the UPAs. The states of the vehicle are defined as $v = \{x, y, s, a\}$, where $x$ and $y$ are the coordinates of vehicles, $s$ represents the velocity of the vehicle, and $a$ denotes the acceleration of the vehicle.

### 3.2 CAST Overview

CAST inpaints the C-V2X scenario by comprehensively analyzing the perceivable information in C-V2X. However, it is difficult to directly utilize the road state in C-V2X due to the sparsity of vehicles with OBU and inconsistent information on the road. Therefore, it is necessary to preprocess the road state, as shown in the top-left of Fig. 1, the preprocess section is divided into two components, which are road gridding and LoS inpainting, respectively. The first component divides the road into occupied grids $G$ that are filled with the states of the road segment. Furthermore, the range of PAs can be first extended to the area in the LoS of vehicles with OBU by broadcasting their sensing information. Thus, the second component pre-inpaints the PAs based on the LoS of the vehicles with OBU. This part will be detailed in Section 4.

Table 2
Notations

| Notation | Description |
|---|---|
| $v$ | States of the vehicle |
| $O$ | Sequence of UPAs |
| $o_i$ | The $i$-th RUPA |
| $Y$ | Real state of the RUPAs |
| $\hat{Y}$ | Generated state of the RUPAs |
| $G$ | Occupied grids |
| $g^j$ | Vector of $j$-th grid cell |
| $g_r^j$ | States of $j$-th grid cell |
| $g_v^j$ | Embedded vector of vehicle state of $j$-th grid cell |
| $U$ | Sequence of RUPAs in this road |
| $u_i$ | The $i$-th RUPA |
| $V_u$ | Number of C-V2X vehicles in each RUPAs |
| $X$ | Temporal road state |
| $X_G$ | Input of GTP |
| $S_o$ | Shallow output |
| $S_h$ | Hidden feature set of shallow output |
| $d_o$ | Output of the model trunk |
| $d_h$ | Hidden features of the model trunk |
| $T_o$ | Output of the big teacher model |

The C-V2X scenario inpainting is carried out to recover the states of vehicles in UPAs in the second section of CAST. Considering inpainting the near UPAs to PAs is easier than the further ones as the influence is more apparent in these UPAs, it is necessary to inpaint UPAs according to their importance one by one. UPAs, however, are irregular in their shape normally, making it difficult to inpaint them sequentially. To this end, we divide the road grid into some regions of equal size, abbreviated as RUPAs, including PAs and UPAs, which are the basic units in inpainting progress. The UPAs to be inpaint in one RUPA would be marked. Then, RUPAs are sorted according to their importance to guide the inpainting process; the RUPA with the highest priority is selected for inpainting in each generation period. The order subsequence of RUPAs is defined as $O = \{o_1, ..., o_i, ..., o_n\}$, where $o_i$ represents the priority of the $i$-th RUPA and $n$ denotes the number of RUPA of the road. Then, the generative model is utilized to inpaint the C-V2X scenario iteratively according to $O$, as shown in Fig. 1, which will be detailed in Section 5. At the end of each inpaint period, the inpainted RUPA is added to the original road grids to help the inpainting of the next period until all RUPAs have been inpainted. In a word, CAST aims to inpaint all of the RUPAs of the road by the known information and the priority sequence of RUPAs, which can be defined as $\hat{Y} = \mathcal{F}(\{G\}_{i=t-w}^{t}, O)$, where $\{G\}_{i=t-w}^{t}$ represents the temporal road state from time $t-w$ to $t$, and $\hat{Y}$ denotes the generated state of the RUPAs at time $t$. The optimization objective of CAST is to precisely inpaint the state of the RUPAs, which can be defined as,

$$\min_{\theta_{\mathcal{F}}} \mathcal{L}(\mathcal{F}(\{G\}_{i=t-w}^{t}, O), Y), \tag{1}$$

where $\theta_{\mathcal{F}}$ represents the parameters of the model of CAST, and $Y$ denotes the real state of the RUPAs at time $t$. $\mathcal{L}$ denotes the deviation between the real state of the RUPAs and the inpainted result generated by CAST. Furthermore, the main symbols are shown in Table 2.

## 4 Data Process

Considering the spatial complexity of UPAs on the road, it is crucial to process data before inpainting the UPAs. In this paper, the road is divided into occupied grids to normalize the road state and transform the original problem into the problem of generating the state of grids to recover the possible vehicle position. In addition, the state of C-V2X vehicles that can be directly sensed by them should be populated in the occupied grids before using the inpainting model.

### 4.1 Road Gridding

This paper represents the road state on occupied grids inspired by image processing, where each grid cell represents the local road state in this scenario, as shown in Fig. 2. It is assumed that $G$ represents the occupied grids with the dimension of $N \times M \times H$, where $N$ denotes the number of grids expanded horizontally by lanes, $M$ is the number of the lane, and $H$ is the dimension of the high-dimensional vector embedded in the grid. In addition, the size of each area of a grid cell in spatial is consistent to normalize the occupied grids, which is defined as $W_{grid} \times L_{grid}$, where $W_{grid}$ and $L_{grid}$ are the width of the grid cell and length of the grid cell, respectively. In this paper, $W_{grid}$ is set to be the width of the lane, and $L_{grid}$ is set to be the length of a normal vehicle.

In this paper, the grid cell is defined as $g^j = \{g_v^j \bigcup g_r^j\}$, where $j$ indicates the $j$-th grid and $g_v^j = \{x, y, s, a\}$ is the embedded vector of vehicle state when there is a car in this grid cell, otherwise $g_v^j = \{x, y, 0, 0\}$. The $g_r^j = \{c, e, q, l, f\}$ represents the state of the $j$-th grid cell, where $c$ indicates whether there is a car in this grid cell. $e$ indicates whether the grid cell is in UPAs. $q$ represents whether the grid cell is in the area to be inpainted in this inpainting period, which is initially set to be 0. $l$ and $f$ are the congestion state of the current lane and the traffic flow of this road, respectively. Thereby, we normalize the road condition and transform the complex brand-new problems into a similar form as the video inpainting problem of GAN by representing the road state with occupied grids.

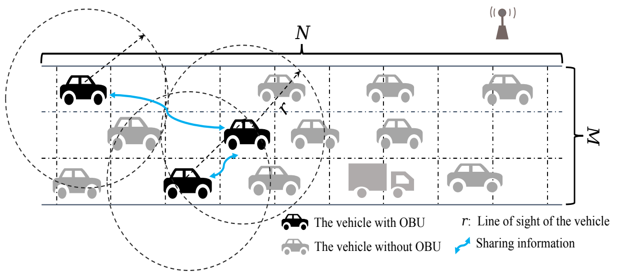


Figure 2. The C-V2X scenario.

### 4.2 LoS Inpainting

First, the states of C-V2X vehicles are filled in the occupied grids for inpainting the UPAs. In this paper, the states of a vehicle $v_x = \{x, y, s, a\}$ are added to the occupied grids according to the coordinates of its center. Thereby, the occupied grids of C-V2X vehicles are obtained. As the sensors of C-V2X vehicles are able to perceive other vehicles within their LoS, as shown in Fig. 2, and the states of these perceived vehicles can be broadcasted to the C-V2X system, these perceived vehicles $v_s = \{x, y, s, a\}$ can also be added to the occupied grids. The LoS of the C-V2X vehicle, however, is not immutable and frozen due to the range of sensors being very vulnerable to the impact of the environment, such as on a foggy day. Thus, we simulate the effect of different weather conditions on the LoS of vehicles by setting various $r$ in the simulation experiment section. In this way, abundant information is utilized in this paper for UPAs inpainting.

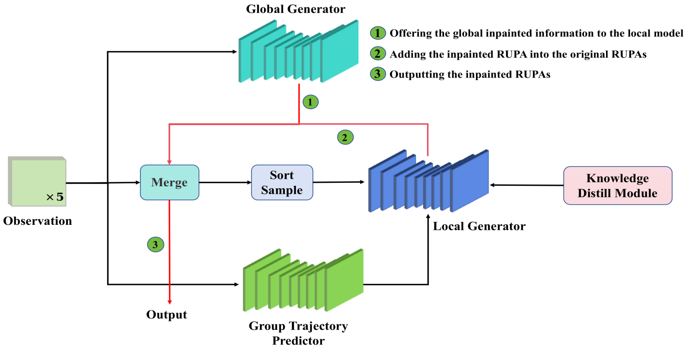

Global Generator
1 Offering the global inpainted information to the local model
2 Adding the inpainted RUPA into the original RUPAs
3 Outputting the inpainted RUPAs
1
2
×5
Observation
Merge
Sort Sample
Local Generator
Knowledge Distill Module
3
Output
Group Trajectory Predictor

Figure 3. The outline model architecture of CAST.

## 5 Model Architecture

As shown in Fig. 3, CAST consists of the following five components, RUPAs sorting algorithm, global generative adversarial model, local generative adversarial model, group trajectory prediction model, and knowledge distillation module, respectively, to precisely and efficiently inpaint the C-V2X scenario. More details about the outline model architecture of CAST are provided in the supplementary file.

Algorithm 1: RUPAs Sorting Algorithm

```
   Input: Occupied grids $G$;
1  Divide $G$ into RUPAs sequence
     $U = \{u_0, ..., u_i, ..., u_n\}$;
2  Initialize $O \leftarrow [\,]$;
3  Initialize $V_u = \{v_0^u, ..., v_i^u, ..., v_n^u\}, \forall\ v_i^u \leftarrow 0$;
4  for i $\leftarrow 0$ to $n$ do
5      $v_i^u \leftarrow$ the number of vehicles in $u_i$
6  end
7  for i $\leftarrow 0$ to $n$ do
8      $index \leftarrow argmax(V_u)$
9      if $index > 1$ then
10         $v_{index-1} \leftarrow v_{index-1} + v_{index}^u$
11     end
12     else
13         $v_{index+1} \leftarrow v_{index+1} + v_{index}^u$
14     end
15     $v_{index} \leftarrow -100$
16     Add $u_{index}$ into $O$
17 end
18 return $O$
```

### 5.1 RUPAs Sorting Algorithm

CAST is used to inpaint the C-V2X scenario for extending the perceivable range of the C-V2X system, namely generating the likely state of UPAs. Inpainting the C-V2X scenario precisely, however, is a challenge due to the intricate interactions of vehicles on the road. Fortunately, there are some intuitive phenomena that can be utilized to simplify our problem. First, the farther away the vehicle is, the less affected it will be by the current vehicle from common sense and vice versa. Then, the influence of the vehicle can be transmitted farther away by the midst vehicle. Take the phenomenon of ghost traffic jams, The influence caused by the deceleration behavior of the vehicle in the front of the lane will be transferred by the vehicle in the middle to the rear of the vehicle in turn. Based on the aforementioned viewpoints, we propose a RUPAs Sorting algorithm, which orders all RUPAs on the road in terms of importance and selects the best UPA successively for inpainting.

The detailed steps of the RUPAs Sorting algorithm are shown in Algorithm 1. First, the road grid is divided into some RUPAs of equal size, which contain some UPAs with irregular shapes. The sequence of RUPAs is defined as $U = \{u_0, ..., u_i, ..., u_n\}$, where $u_i$ represents the $i$-th RUPA, and $n$ is the number of RUPAs in the occupied grids. The task of C-V2X scenario inpainting, thereby, is decomposed into multiple sub-tasks of RUPA inpainting. After that, the priority of each RUPA can be computed by its importance. The importance of each unit in the occupied grids is interrelated to the distance from other vehicles equipped with OBU according to the previous point of view. Thus, we suppose that the initial priority of each RUPA is represented by the number of C-V2X vehicles in this RUPA, and then dynamically adjust the priority of each RUPA to simplify the computing process. The initial priority sequence of RUPAs is set to be an empty list, as shown in Line 2 in Algorithm 1. The list of the number of C-V2X vehicles in each RUPA is defined as $V_u = \{v_0^u, ..., v_i^u, ..., v_n^u\}$, where $v_i^u$ is the number of C-V2X vehicle in $u_i$. In each process of Lines 9-20, we always select the RUPA with the highest $v^u$, then the $v^u$ of its neighbor RUPA is adjusted by its $v^u$ to transfer its importance. Finally, the priority of each RUPA for inpainting $O = \{o_1, ..., o_i, ..., o_n\}$ is obtained.

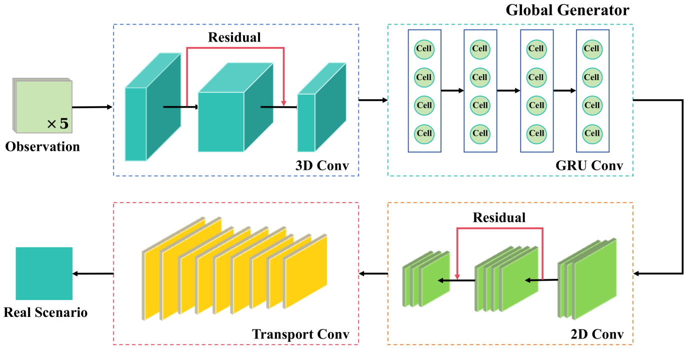


Figure 4. The architecture of the global generator in CAST.

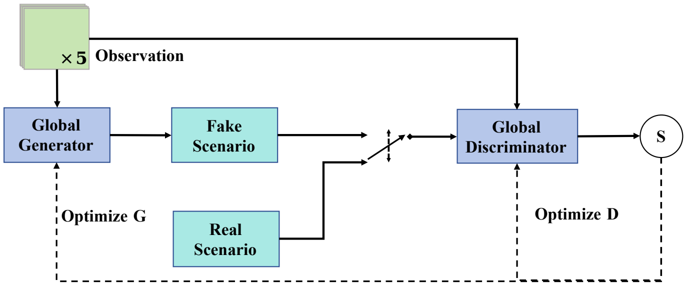


Figure 5. The overall architecture of the global generative adversarial model in CAST.

### 5.2 Global Generative Adversarial Model

The global generative adversarial model aims to pre-inpaint the C-V2X scenario before inpaint each RUPA according to its priority by providing a global sight. In this way, CAST will not over-focus on RUPA with higher priority. As shown in Fig. 4, the global generator does not take the local features of a single RUPA into account, but the global features are only considered to inpaint the whole gridded road state. Thus, it does not consider the priority order of RUPAs. First, the global observation $X = \{G^i\}_{i=t-w}^{t}$ passes the 3D convolution layers; its temporal-spatial interaction information is extracted. Then, the extracted representation is sent to the Gate Recurrent Unit (GRU) convolution layer to further extract its temporal-spatial interaction information. The expression functions of each GRU cell are defined as,

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}), \tag{2}$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}), \tag{3}$$

$$h_t^{'} = \tanh(Wx_t + r_t \odot Uh_{t-1}), \tag{4}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t^{'}, \tag{5}$$

where $z_t$ denotes the update gate, and $r_t$ represents the reset gate. $h_t^{'}$ is the current memory content, and $h_t$ is the final memory at the current time step.

The global generative adversarial architecture is shown in Fig. 5. More details about this architecture are provided in the supplementary file. In this paper, we adopt Wasserstein GAN-gradient penalty (WGAN-GP) [33, 34] as the training framework. In this way, the discriminator will score the samples of the unperceived RUPAs to express how real the input sample is. Specifically, the score by the global discriminator can be defined as $b^{global} = \mathcal{D}(X, Y_d)$, where $Y_d$ represents the real or generated road state of the RUPAs. The objective loss functions of the global inpainting module are defined as,

$$\begin{aligned}\mathcal{L}_d^{global} &= \mathbb{E}_{x\sim P_g^{global}}[b^{global}] - \mathbb{E}_{x\sim P_r}[b^{global}] + \\ &\qquad \lambda \mathbb{E}_{x\sim\chi}[|| \bigtriangledown_x b^{global}||_p - 1],\end{aligned} \tag{6}$$

$$\mathcal{L}_g^{global} = -\mathbb{E}_{x\sim P_g^{global}}[b^{global}]. \tag{7}$$

In addition to the adversarial loss, we also design a regularization term for the global generator to guide the training process of the global generator. The regularization term is defined as follows,

$$\mathcal{L}_{re}^{global} = ||Y^{global} - \hat{Y}^{global}||_2^2. \tag{8}$$

Therefore, the overall loss function of the global generative adversarial model is defined as,

$$\mathcal{L}_{global} = \mathcal{L}_d^{global} + \mathcal{L}_g^{global} + \mathcal{L}_{re}^{global}. \tag{9}$$

## 5.3 Local Generative Adversarial Architecture

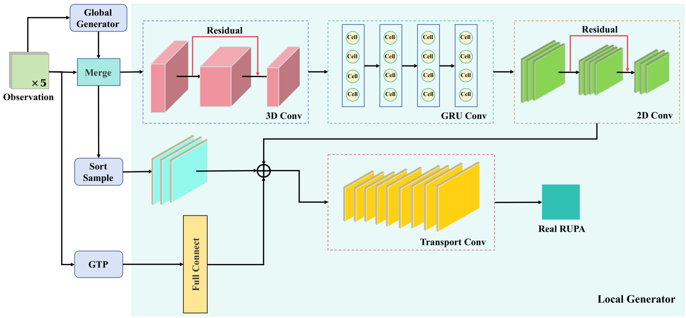


Figure 6. The architecture of the local generator in CAST.

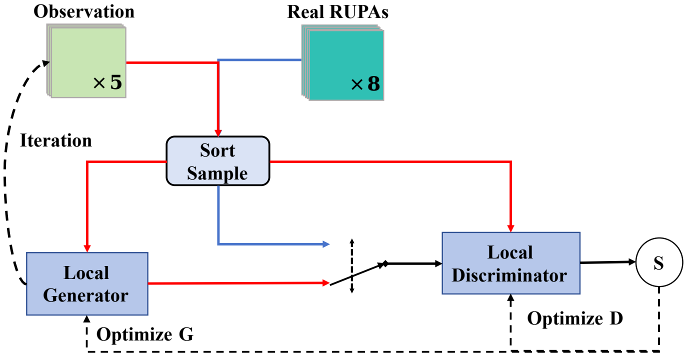


Figure 7. The overall architecture of the local generative adversarial model in CAST.

Considering that inpainting the UPAs closer to PAs is easier than the further ones, this is due to the fact that the interaction between the vehicles in the nearer UPAs and PAs is more remarkable than in the more distant UPAs, it is necessary to preferentially inpaint the near UPAs, namely, the RUPAs with higher priority. In the local generator, we design an iterative inpainting strategy according to the priority sequence of the RUPAs to take full advantage of the inpainted information of the RUPAs with higher priorities. First, the pre-inapinted result from the global generator is also merged into the input of the local generator as Step 1 in Fig. 3. In the iterative inpainting strategy, the RUPA with the highest priority will be inpainted first by the local generator, and then the inpainted RUPA will be merged with the current input of the local generator as Step 2 in Fig. 3. Finally, the inpainting process will end when the RUPA with the lowest priority has been inpainted, as shown in Step 3 in Fig. 3, and the inpainted result will be output.

As shown in Fig. 6, the local generator uses 3D convolution layers, GRU convolution layers, and 2D convolution layers to extract the temporal-spatial interaction information. The state of the target RUPA that needs to be inpainted in this inpainting period is utilized to help the local generator focus on this RUPA, and the features of this RUPA are extracted by the 2D convolutions. Moreover, given that the influence of interaction between vehicles has a certain hysteresis, reconstructing the current C-V2X scenario by analyzing the historical state of the C-V2X scenario is a feasible way to further extract the interaction between PAs and UPAs. Therefore, the group vehicle prediction-based check module is designed to further extract this interaction. The details of the group vehicle trajectory prediction model are discussed in the next section. Then, all the extracted information is added. Finally, the inpainted RUPA is obtained by the transport convolution layers.

The local generative adversarial architecture is shown in Fig. 7. More details of this architecture are provided in the supplementary file. The objective loss functions of the local inpainting module are defined as,

$$\begin{aligned}\mathcal{L}_d^{local} &= \mathbb{E}_{x\sim P_g^{local}}[b^{local}] - \mathbb{E}_{x\sim P_r}[b^{local}] + \\ &\qquad \lambda \mathbb{E}_{x\sim\chi}[|| \bigtriangledown_x b^{local}||_p - 1],\end{aligned} \tag{10}$$

$$\mathcal{L}_g^{local} = -\mathbb{E}_{x\sim P_g^{local}}[b^{local}]. \tag{11}$$

Aside from the adversarial loss function, we also add a regularization term for the local generator to guide the training process of the local generator. The regularization term is defined as follows,

$$\mathcal{L}_{re}^{local} = \frac{1}{n}\sum_{i=1}^{n} ||Y_i^{local} - \hat{Y}_i^{local}||_2^2. \tag{12}$$

Thus, the overall loss function of the local generative adversarial model is defined as,

$$\mathcal{L}_{local} = \mathcal{L}_d^{local} + \mathcal{L}_g^{local} + \mathcal{L}_{re}^{local}. \tag{13}$$

## 5.4 Group Vehicle Trajectory Prediction Model

Given the lag nature of vehicular interactions, we propose a group vehicle trajectory prediction model to further extract this interaction between UPAs and PAs, namely GTP. More details about GTP are provided in the supplementary file.

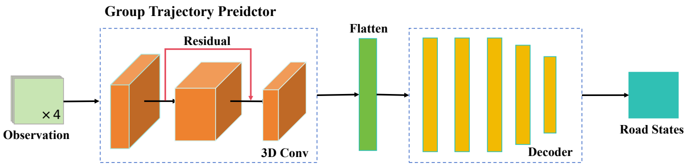


Figure 8. The architecture of the group trajectory prediction model in CAST.

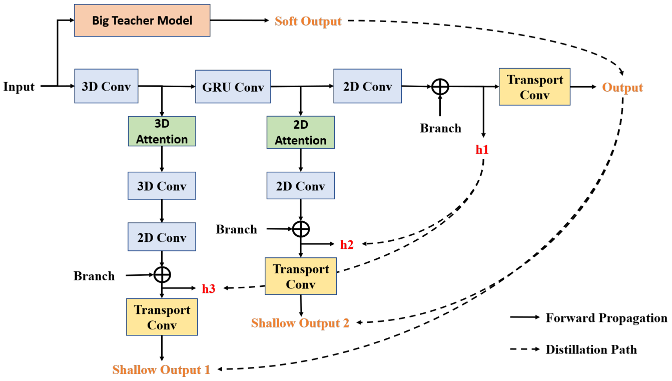


Figure 9. The architecture of self-distillation and cross-model distillation.

### 5.5 Knowledge Distillation

Although the above model can precisely inpaint the RUPAs, its low computation efficiency makes it unbearable for some computation-limited C-V2X infrastructures when deploying this model. Therefore, it is essential to ensure the computation efficiency and the inpainting precision of the model before deploying it to the real C-V2X scenario. To date, the KD technology has been extensively verified in a model compressor, and it can obtain an excellent tradeoff between efficiency and precision. Thus, inspired by [35], we design a KD-based architecture to simultaneously improve computation efficiency and inpainting accuracy. As shown in Fig. 9, we combine the self-distillation and cross-model distillation frameworks to optimize the model ready for deployment, which contains two shallow inpainting branches for generating shallow inpainting output, an inpainting trunk for generating deep inpainting output, and a big teacher model for transferring its dark knowledge to the model ready for deployment. This architecture is suitable for the promising cloud-edge framework, in which the big teacher model can be trained and deployed in the cloud equipped with sufficient data and computation capability, and the tailored model that will be deployed at the edge can be trained by the knowledge from the big teacher model and the self-distillation. This tailored model could be a personalized model for a specific scenario. We suppose that the shallow output set is defined as $S_o = \{s_o^1, ..., s_o^K\}$, where $K = 2$ in this architecture, and the hidden feature set of shallow branches is defined as $S_h = \{s_h^1, ..., s_h^K\}$. The output and hidden features of the model trunk are defined as $d_o$ and $d_h$, respectively. The output of the big teacher model is defined as $T_o$. As shown in Fig. 9, we design some distillation paths to transfer knowledge in this architecture. These distillation paths are divided into two classes, the self-distillation path, and the cross-model distillation path. The self-distillation loss is defined as,

$$\mathcal{L}_{dis}^s = \frac{1}{K}\sum_{k=1}^{K}\Big((1-\alpha)\mathcal{L}_{BCE}(s_o^k, r) + \alpha\mathcal{L}_{KL}(s_o^k, d_o) + \lambda\mathcal{L}_2(s_h^k, d_h)\Big), \tag{14}$$

where $\mathcal{L}_{BCE}$ is the binary entropy loss, $\mathcal{L}_{KL}$ is the Kullback-Leibler divergence, $\mathcal{L}_2$ denotes the mean square error. The cross-model distillation loss is defined as,

$$\mathcal{L}_{dis}^c = \mathcal{L}_{KL}(d_o, T_o). \tag{15}$$

A brief rationale for the effectiveness of the knowledge distillation model in improving the training process of the C-V2X scenario inpainting model is provided in the supplementary file.

Finally, the inpainting model trained by distillation architecture also considers the adversarial loss. Therefore, the overall objective loss function of CAST can be expressed as,

$$\mathcal{L}_{CAST} = \mathcal{L}_{global} + \mathcal{L}_{local} + \mathcal{L}_{dis}^s + \mathcal{L}_{dis}^c. \tag{16}$$

## 6 Simulation and Performance Evaluation

This section conducts the performance evaluation of our proposed effiCient trAffic Scenario inpainTing (CAST). First, we introduce the experimental environment. Second, the evaluation metrics proposed in this paper for the brand-new C-V2X scenario inpainting problem are detailed. Finally, the experimental simulations are conducted to evaluate the performance of CAST.

### 6.1 Parameter Settings

To validate the performance of inpainting the C-V2X scenario of CAST, we implement the simulation based on the traffic data of the HighD dataset (real-world data) [36] and SUMO-DS (synthetic data). SUMO-DS generated from SUMO tool [37]. More details on dataset are provided in the supplementary file. The size of the road network of SUMO-DS is set to $1500m \times 1500m$; this road network in a grid shape contains 16 junctions and 40 roads. There are 700 to 900 vehicles on the road network in SUMO-DS on average per second. Then, the C-V2X scenario is simulated by these road data. In this simulation, the length of each road segment is set to be 300m, and the probability of a vehicle equipped with an OBU device and the maximum distance of the LoS of each C-V2X vehicle are both variables.

We use Python 3.8 to implement CAST, and the model of CAST is built by Pytorch 1.12.1. To optimize the proposed model, we use Adam as the optimizer. The learning rate and factor decay of training are assigned as 0.0004 and 0.5, respectively. The batch size and trained epochs are set to 512 and 1000, respectively. All experiments are run on the machine with Intel Xeon Gold 6226R CPU@2.90GHZ with 128GB memory. In addition, more simulation parameters are shown in Table 3. The source code and the dataset of this paper are available at link[1].

[1] https://github.com/NetworkCommunication/PEER

Table 3
Simulation parameters

| Parameter | Value |
|---|---|
| Network size of SUMO | $1500m \times 1500m$ |
| Number of junctions of SUMO | 16 |
| Number of roads of SUMO | 40 |
| Number of vehicles of SUMO | 700∼900 |
| Vehicle speed of SUMO | 0∼30m/s |
| Length of the road segment | 300m |
| Learning rate | 0.0004 |
| Factor decay | 0.5 |
| Batch size | 512 |
| Epochs | 250 |
| Proportion of C-V2X vehicle | 0.1, 0.2, 0.3, 0.4, 0.5 |
| Distance of LoS of C-V2X vehicle | 20, 30, 40, 50, 60, 80, 100 |

### 6.2 Evaluation Metrics

In light that the C-V2X scenario inpainting is a brand-new initiative and the evaluation metrics of both the vehicle trajectory prediction and video inpainting fields are unable to measure the inpainting performance, we propose four dedicated evaluation metrics to comprehensively evaluate the performance of CAST for the C-V2X scenario inpainting.

1) Precise Inpaint Rate (PIR): PIR is the high-precise C-V2X scenario inpainting rate, which represents the proportions of perfectly inpainted vehicles to measure the precise inpainting completeness. If the grid of an inpained vehicle contains a real vehicle, it is called a perfect inpainted case. The bigger the value, the better the result is for safe applications. The PIR is defined as,

$$PIR = \frac{Z_o + Z_p}{Z}, \tag{17}$$

where $Z$ is the total number of vehicles, including observed and unobserved vehicles, on the road. $Z_o$ denotes the number of vehicles in PAs. $Z_p$ is the number of vehicles that are perfectly inpainted.

2) Rough Inpaint Rate (RIR): Although PIR can measure perfect inpainting completeness, the deviation between the inpainted vehicular position and the real vehicular position cannot be computed. Thus, we propose the RIR to evaluate the rough performance of CAST, which measures the inpainting effectiveness by calculating the ratio of the number of inpainted vehicles to the number of real vehicles in each RUPA. The bigger the value, the better the result is for the applications that require the number of vehicles at the RUPA level. The RIR is defined as,

$$RIR = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}\frac{s_i^{'}}{s_i}, & \frac{s_i^{'}}{s_i} \leq 1\\ \frac{s_i}{s_i^{'}}, & \frac{s_i^{'}}{s_i} > 1,\end{cases} \tag{18}$$

where $n$ represents the number of RUPA on this road. $s_i^{'}$ represent the number of vehicles in the inpainted UPAs and PAs on the area of the $i$-th RUPA. $s_i$ is the number of both observed and unobserved vehicles on the area of the $i$-th RUPA in the real scenario.

3) Inpaint Confidence Error (ICE): It is unrealistic to guarantee that each grid in the inpainted C-V2X scenario determines whether it contains a vehicle. Thus, ICE is proposed to measure the confidence error of each grid as to whether it contains vehicles. The smaller the values are, the better the result is. The ICE is defined as,

$$ICE = \frac{1}{N \times M}\sum_{i=1}^{N \times M}\sqrt{(\hat{p_i} - p_i)^2}, \tag{19}$$

where $\hat{p_i}$ and $p_i$ represent the confidence of whether there exists a vehicle in the $i$-th inpainted and real grid, respectively.

4) Lane Level Inpaint Rate (LLIR): Given that lane-level inpainting is more necessary in some situations, e.g., making a lane-changing strategy. Therefore, we propose the LLIR to evaluate the performance of CAST at the lane level by calculating the ratio of inpainted vehicle numbers to real vehicle numbers in each lane in RUPAs. The bigger the value, the better the result is for applications that need lane-level vehicle information. The LLIR is defined as,

$$LLIR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{M}\sum_{j=1}^{M}\begin{cases}\frac{s_{i,j}^{'}}{s_{i,j}}, & \frac{s_{i,j}^{'}}{s_{i,j}} \leq 1\\ \frac{s_{i,j}}{s_{i,j}^{'}}, & \frac{s_{i,j}^{'}}{s_{i,j}} > 1\end{cases} \tag{20}$$

where $M$ represents the lane number on the road. $s_{i,j}^{'}$ represent the number of vehicles in the $j$-th lane of inpainted UPAs and PAs on the area of the $i$-th RUPA. $s_{i,j}$ is the number of both observed and unobserved vehicles in the $j$-th lane on the area of the $i$-th RUPA in the real scenario

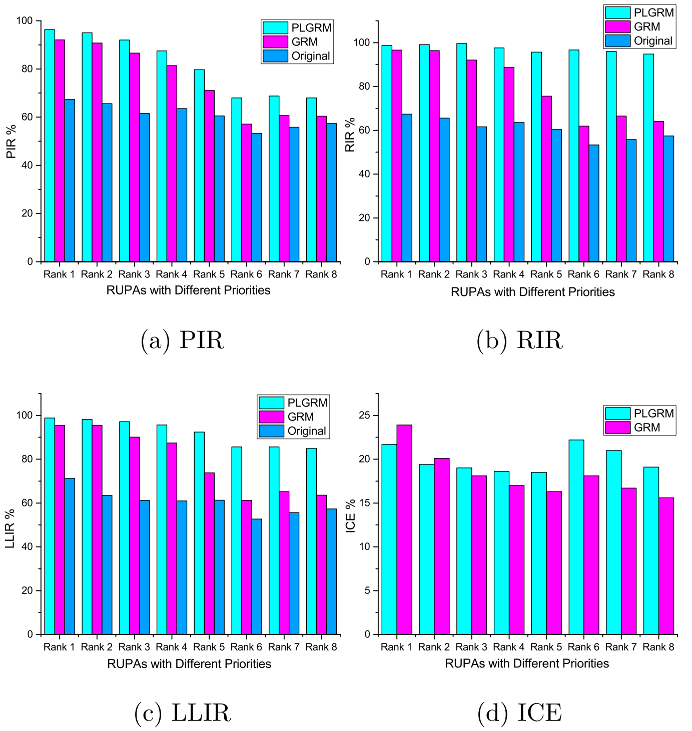
In this way, the performance of CAST can be comprehensively evaluated by utilizing the aforementioned evaluation metrics to empower C-V2X applications.

Table 4
Ablation study results

| Measures | PLGRM | LRM | LGRM | GRM |
|---|---|---|---|---|
| PIR | 0.80 | 0.77 | 0.78 | 0.75 |
| RIR | 0.97 | 0.91 | 0.96 | 0.79 |
| LLIR | 0.90 | 0.80 | 0.90 | 0.62 |
| ICE | 0.25 | 0.27 | 0.25 | 0.24 |

### 6.3 Experimental Results

CAST mainly consists of three components, RUPAs sorting algorithm, generative adversarial construction, and knowledge distillation, respectively. In this section, the performance of the proposed generative model is evaluated first. The generative model consists of the group Prediction module, the Local inpainting module, and the Global inpainting module, respectively. So we call it PLGRM (PLG geneRative Model). Thus, the ablation experiment of PLGRM is conducted first to verify the effectiveness of each component in PLGRM. Then, the performance of PLGRM in variable scenarios is demonstrated. Moreover, the feasibility of the knowledge distillation module is also evaluated. Thus, we finally evaluate the overall performance of CAST, including PLGRM and knowledge distillation. To be more specific, we compared four models, 1) the small PLGRM with a small number of parameters,



(a) PIR

(b) RIR

(c) LLIR

(d) ICE

Figure 10. Inpainted results of RUPAs with different priorities, where (a) represents the PIR of different RUPAs, (b) represents the RIR of different RUPAs, (c) represents the LLIR of different RUPAs, and (d) represents the ICE of different RUPAs.

which is denoted by the small model; 2) the big PLGRM with enormous parameters, denoted by the self-distillation; 3) the small PLGRM with only self-distillation, denoted by the teacher model; and 4) the small PLGRM with both self-distillation and cross-model distillation from the big PLGRM, namely CAST. In addition, we also discuss other benefits of knowledge distillation in this part.

### 6.3.1 Ablation Experiment of PLGRM

PLGRM mainly consists of three components, the group prediction module, the global inpainting module, and the local inpainting module, respectively. In order to investigate the performance of each component of PLGRM, this section conducts the ablation experiment on the SUMO dataset. In this section, we compare the performance of 1) the entire generative adversarial model (PLGRM), 2) Local geneRative Model (LRM), 3) Global geneRative Model (GRM), and 4) Local-Global generative Model (LGRM), in terms of four measures introduced above. The result of the ablation experiment can be seen in Table 4, the rate of the C-V2X vehicle is 10% and the distance of the line of sight is 50m in this ablation experiment.

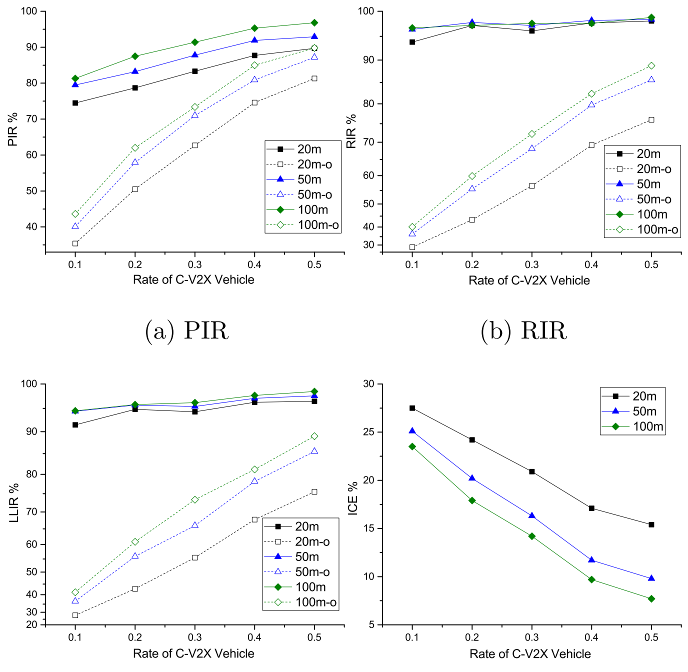
As shown in Table 4, the GRM performs worst in PIR, RIR, and LLIR, in which PIR is 75% near the performance of other modules, however, RIR and LLIR of it are far less than those of other modules, which indicates the global inpainting model is weak in precise inpainting, leading to an inpainted vehicle number that is higher than the real scenario. Moreover, the best performance of GRM in ICE indicates that the global inpainting module is more justice to different RUPAs with diverse priorities. Compared with GRM, LRM performs better in terms of PIR, RIR, and LLIR but worse in ICE, which indicates the local inpainting model is better at precise inpainting. However, the local inpainting is biased towards the RUPAs with high priority due to the accumulated errors of the RUPAs with lower priority. LGRM performs better in all four metrics compared with LRM, especially in terms of the RIR, LLIR, and ICE, which indicates that the local-global inpainting architecture possesses the advantages of both the local and global inpainting models, namely inpainting accuracy and justice for RUPAs with different priorities. PLGRM simultaneously consists of the local and global inpainting model and the group prediction model, which performs best in terms of PIR, RIR, and LLIR compared with LRM, LGRM, and GRM, which demonstrates that the predicted information generated by the group prediction model is valuable for inpainting a more accurate C-V2X scenario. Therefore, we can conclude that each component in PLGRM is indispensable for precisely inpainting C-V2X scenarios.

### 6.3.2 Inpainted Results of PLGRM

In this section, we utilize the proportion of C-V2X vehicles on the road to represent the development stage of the C-V2X for simulating the future practical C-V2X scenario. Meanwhile, we represent the effect of climate by the distance of the LoS of the sensor in the vehicle. Then, we comprehensively evaluate the performance of PLGRM in these scenarios based on the SUMO and HighD datasets, respectively, in terms of four measures, namely PIR, RIR, LLIR, and ICE.

First, we explore the inpainting results of each RUPA with different priorities based on PLGRM and GRM to further discuss the limitations of our proposed method in the setting that the rate of the C-V2X vehicle is 20% and the distance of LoS is 80m. As shown in Fig. 10, in the original scenario that only contains vehicles with OBUs and in the LoS of the C-V2X system, the proportions of RUPAs with different priorities in terms of PIR, RIR, and LLIR are only slightly different. However, if only the global generative model were utilized, the difference in the inpainting performance of different RUPAs would increase. Fortunately, PLGRM mitigates the difference in the inpainting outcome between different RUPAs, especially in terms of RIR and LLIR. Therefore, although the precise inpainting results, PIR, of RUPAs with lower priorities might be unable to meet the needs of applications requiring high precision, the needs of applications with RUPA-level and lane-level requirements are ensured.

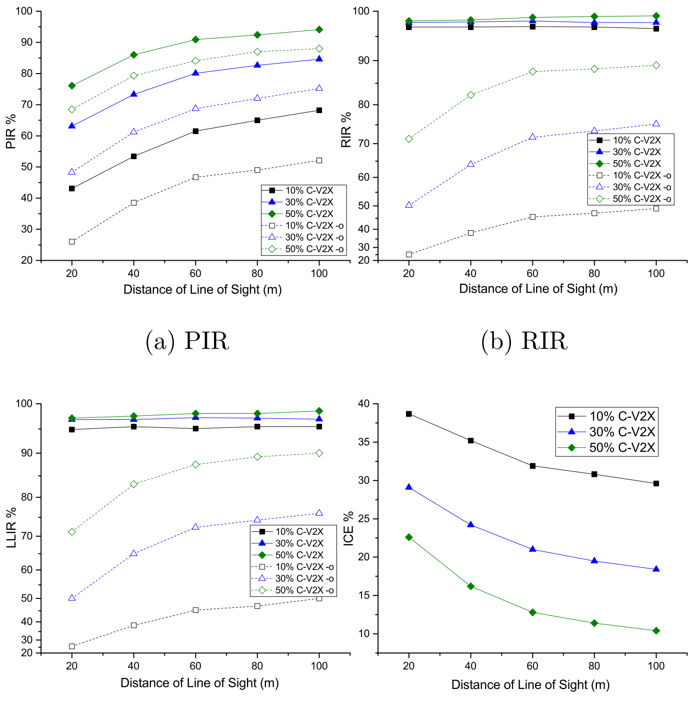
Furthermore, as shown in Fig. 11, we test PLGRM in 15 scenarios based on SUMO datasets, in which the proportions of C-V2X vehicles are set to 10%, 20%, 30%, 40%, and 50%, respectively, and the distances of the LoS of the vehicle are set to 20 m, 50 m, and 100 m, respectively. The proportions of the original scenarios, only containing vehicles with OBUs and other vehicles in the LoS of the C-V2X system, in terms of PIR, RIR, and LLIR, are also offered to demonstrate the effectiveness of the proposed method. We add $-o$ to denote these scenarios in Fig. 11. It can be seen that PIR of PLGRM is close to 75% when the proportion of C-V2X vehicles and the distance of the LoS of the vehicle are 10% and 20 m, respectively, and PIR is over 96% when they are respectively 50% and 100 m. The growth of PIR can be seen when the C-V2X vehicle perceives a more extensive area, and accordingly, ICE decreases, which means that the inpainted



(a) PIR (b) RIR

(c) LLIR (d) ICE

Figure 11. Inpainted results of PLGRM based on the SUMO dataset for different measures, where (a) represents the PIR of PLGRM, (b) represents the RIR of PLGRM, (c) represents the LLIR of PLGRM, and (d) represents the ICE of PLGRM.



(a) PIR (b) RIR

(c) LLIR (d) ICE

Figure 12. Inpainted results of PLGRM based on the HighD dataset for different measures, where (a) represents the PIR of PLGRM, (b) represents the RIR of PLGRM, (c) represents the LLIR of PLGRM, and (d) represents the Con of PLGRM.

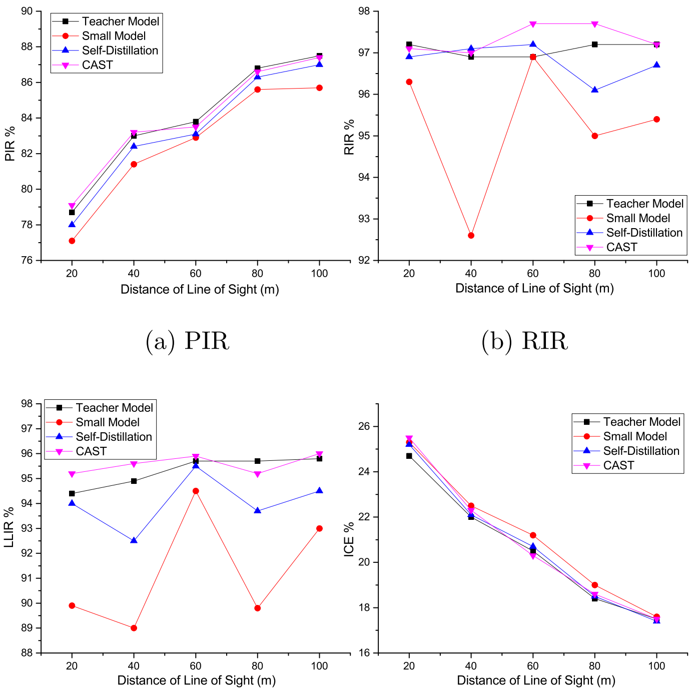
result is more credible. Furthermore, RIR and LLIR represent the inpainting effect at the RUPA level and lane level, which reflects the redrawing deviation degree of the model redrawing to a certain extent. As shown in Fig. 11(b), RIR is over 90% in each scenario and near 99% when the proportion of C-V2X vehicles and the distance of the LoS of the vehicle are 50% and 100m, respectively. It illustrates that the majority of the inpainted vehicles by PLGRM will not deviate from their RUPA. Thus, the inpainted result of PLGRM can be used in some applications that rely on global road information. As shown in Fig. 11(c), the LLIR is over 86% in each scenario and reaches 98% in the best scenario. It demonstrates that the majority of the inpainted vehicles by PLGRM will not deviate from their lane. Thus, the inpainted result of PLGRM can be used in some applications that require lane-level global sight.

In order to validate the effectiveness of our proposed method in the real road scenario, we evaluate PLGRM once again in the real road dataset. As shown in Fig. 12, we test PLGRM in 15 scenarios based on the HighD datasets, in which the proportions of C-V2X vehicles are set to 10%, 30%, and 50%, respectively, and the distances of the LoS of the vehicle are set to 20 m, 40 m, 60 m, 80 m, and 100 m, respectively. The proportions of the original scenarios, in terms of PIR, RIR, and LLIR, are also offered in this setting. We add $-o$ to denote these scenarios in Fig. 12. As shown in Fig. 12(a), the performance of PLGRM in terms of PIR is not excellent as its performance in the SUMO dataset in the worst situation. This is because the road condition in the HighD dataset is more complex and diverse than that in the SUMO dataset. For all that, PIR of PLGRM is over 90% when the proportion of C-V2X vehicles and the distance of the LoS are over 50% and 60 m, respectively. In addition, ICE is higher when the range of perception of the C-V2X system is larger, as shown in Fig. 12(d). Although the performance of PLGRM is not always good in each scenario in the HighD dataset, the performance of PLGRM in terms of RIR and LLIR is excellent in each scenario. As shown in Fig. 12(b) and (c), RIR and LLIR of PLGRM all exceed 95% in the worst situation and exceed 99% when the proportion of C-V2X vehicles and the distance of the LoS are 50% and 100 m, respectively, which indicate the superior performance of PLGRM in the inpainting task at RUPA-level or lane-level.

We verify the effectiveness of PLGRM in the C-V2X scenario inpainting task by experimenting with our simulation platform based on the SUMO and HighD datasets. Given the aforementioned analysis of the experiment results, we found that the PLGRM performs excellently at the inpainting task at RUPA-level and lane-level in each scenario, but the inpainting performance at a precise level is susceptible to the complexity of the road environment. Thus, the inpainted result is useful for some C-V2X applications that require RUPA-level or lane-level global information, even if only 10% of vehicles are equipped with C-V2X devices. As for some C-V2X applications that require precise global information, the inpainted result by PLGRM is useful when the proportion of C-V2X reaches 50%.

### 6.3.3 Inpainted Results of Overall CAST

In this section, we assess the overall performance of CAST, including both PLGRM and the knowledge distillation architecture, to verify the effectiveness of the knowledge distillation



(a) PIR

(b) RIR

(c) LLIR

(d) ICE

Figure 13. Inpainted results of different models based on the SUMO dataset for different measures, where (a) represents the performance in terms of PIR, (b) represents the performance in terms of RIR, (c) represents the performance in terms of LLIR, and (d) represents the performance in terms of ICE.

architecture in increasing computation efficiency and adapting to various computation-limited C-V2X infrastructures. As shown in Fig. 13, we compare the performance of four models, 1) a small PLGRM with 0.55 million parameters, called the Small Model; 2) a small PLGRM with 0.55 million parameters only using self-distillation, which is denoted by self-distillation; 3) a small PLGRM with 0.55 million parameters using both self-distillation and cross-model distillation, namely CAST; and 4) a big PLGRM with 1.55 million parameters, namely the Teacher Model, in the scenario where the proportion of vehicles with OBUs is set to 20%.

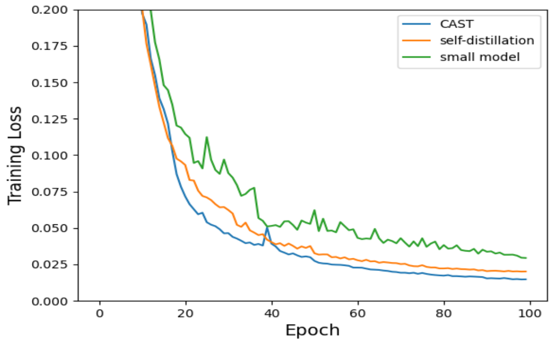


Figure 14. The training process.

As shown in Fig. 13, the small model performs the worst in each measure due to its low learning ability. The performance of the small model based on self-distillation is much better than the small model and sometimes even higher than the performance of the big teacher model, which proves that the self-distillation module greatly improves learning ability. In most cases, the performance of the CAST is near or exceeds the performance of the big-teacher model, and it is better than the small model based on self-distillation, which indicates the effectiveness of combining self-distillation and cross-model distillation to further increase the learning ability of the C-V2X scenario inpainting model. Furthermore, the training errors of the small model, the small model based on self-distillation, and CAST are shown in Fig. 14, in which CAST has the fastest convergence speed than the small model and the small model based on self-distillation, and the small model has the slowest convergence. Thus, the knowledge distillation architecture remarkably accelerates the convergence speed when training the C-V2X scenario inpainting model.

Table 5
Self-distillation results

| Measures | 0.55M | 0.86M | 0.98M | 1.55M(PLGRM) |
|---|---|---|---|---|
| PIR | 86 | 86.2 | 86.8 | 86.8 |
| RIR | 96.1 | 97.4 | 97.2 | 97.2 |
| LLIR | 93.7 | 94.5 | 95 | 95.7 |
| ICE | 20.5 | 20.7 | 19.8 | 19.9 |

In the absence of the teacher model, the self-distillation architecture can be used to simultaneously ensure the inpainting precision and efficiency. Thus, we investigate the performance of different sizes of models based on self-distillation on the setting that the rate of C-V2X vehicle is 20% and the distance of LoS is 80m. As shown in Table 5, the performance of the model with 0.98 million parameters is near the performance of the big model. Therefore, the compression efficiency of CAST is 63% even without the knowledge from the big teacher model, and it will reach 35% if it utilizes the knowledge from the big teacher model. Thus, CAST is useful to guarantee inpainting precision and computation efficiency when taking diverse computation-limited devices into account when deploying the C-V2X inpainting model.

## 7 Conclusion

This paper pionners to tackle the challenging inpainting problem in the C-V2X scenario. Specifically, we propose an efficient C-V2X scenario inpainting method, CAST, which aims to generate the state of the unperceived area on the road. Extensive experiments are conducted to verify the outstanding performance of CAST in the inpainting task of the C-V2X scenario. In our future work, we plan to continue our research further on inpainting to study the directions, e.g., generating more diverse information about the vehicle with considering more complex road scenarios. As a concluding remark, we expect that the proposed method can also be extended to solve the inpainting problem in different vehicular communication scenarios under a wide range of applications. This will be a concrete step towards interesting research lines and horizons.

## Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant 62372310, and in part by the Liaoning Province Applied Basic Research Program under Grant 2023JH2/101300194, and in part by the LiaoNing Revitalization Talents Program under Grant XLYC2203151.

## References

[1] W. Feng, S. Lin, N. Zhang, G. Wang, B. Ai, and L. Cai, “Joint C-V2X based offloading and resource allocation in multi-tier vehicular edge computing system,” IEEE Journal on Selected Areas in Communications, vol. 41, no. 2, pp. 432–445, 2023.

[2] C. Guo, C. Wang, L. Cui, Q. Zhou, and J. Li, “Radio resource management for C-V2X: From a hybrid centralized-distributed scheme to a distributed scheme,” IEEE Journal on Selected Areas in Communications, vol. 41, no. 4, pp. 1023–1034, 2023.

[3] K. Sehla, T. M. T. Nguyen, G. Pujolle, and P. B. Velloso, “Resource allocation modes in C-V2X: From lte-V2X to 5g-V2X,” IEEE Internet of Things Journal, vol. 9, no. 11, pp. 8291–8314, 2022.

[4] T. G. and R. H., “Vehicle-to-nothing? securing C-V2X against protocol-aware dos attacks,” in Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications, 2022, pp. 1629–1638.

[5] L. Chia-Hung, L. Shih-Chun, W. Chien-Yuan, and C. Thomas, “A C-V2X platform using transportation data and spectrum-aware sidelink access,” in Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021, pp. 1293–1298.

[6] A. Bazzi, C. Campolo, B. M. Masini, and A. Molinaro, “How to deal with data hungry V2X applications,” in Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, 2020, pp. 333–338.

[7] P. Wang, B. Di, H. Zhang, K. Bian, and L. Song, “Platoon cooperation in cellular V2X networks for 5g and beyond,” IEEE Transactions on Wireless Communications, vol. 18, no. 8, pp. 3919–3932, 2019.

[8] C. Qu, W. Y. Qi, and P. Wu, “A high precision and efficient time-to-collision algorithm for collision warning based V2X applications,” in Proceedings of the 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS), 2018, pp. 1–5.

[9] M. C. Lucas-Esta, B. Coll-Perales, T. Shimizu, J. Gozalvez, T. Higuchi, S. Avedisov, O. Altintas, and M. Sepulcre, “An analytical latency model and evaluation of the capacity of 5g nr to support V2X services using v2n2v communications,” IEEE Transactions on Vehicular Technology, vol. 72, no. 2, pp. 2293–2306, 2022.

[10] C. Wang, H. Huang, X. Han, and J. Wang, “Video inpainting by jointly learning temporal structure and spatial details,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 5232–5239.

[11] L. Zhao, Y. Liu, A. Al-Dubai, A. Zomaya, and A. Hawbani, “A novel generation-adversarial-network-based vehicle trajectory prediction method for intelligent vehicular networks,” IEEE Internet of Things Journal, vol. 8, no. 3, pp. 2066–2077, 2021.

[12] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, 1999, pp. 1033–1038.

[13] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, “Free-form image inpainting with gated convolution,” in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 4471–4480.

[14] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” ACM Transactions on Graphics (ToG), vol. 36, no. 4, pp. 1–14, 2017.

[15] Y. Wexler, E. Shechtman, and M. Irani, “Space-time video completion,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. I–I.

[16] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, “Video inpainting of complex scenes,” Siam Journal on Imaging Sciences, vol. 7, no. 4, pp. 1993–2019, 2014.

[17] Z. Li, C. Z. Lu, J. Qin, C. L. Guo, and M. M. Cheng, “Towards an end-to-end framework for flow-guided video inpainting,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17 562–17 571.

[18] Y. Wu, V. Singh, and A. Kapoor, “From image to video face inpainting: spatial-temporal nested GAN (STN-GAN) for usability recovery,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2396–2405.

[19] X. Zou, L. Yang, D. Liu, and Y. J. Lee, “Progressive temporal feature alignment network for video inpainting,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16 448–16 457.

[20] L. Ke, Y. Tai, and C. Tang, “Occlusion-aware video object inpainting,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14 468–14 478.

[21] N. Nguyen, B. Pham, T. Thai, and M. Nguyen, “An improved GAN-based approach for image inpainting,” in Proceedings of the 2021 RIVF International Conference on Computing and Communication Technologies (RIVF), 2021, pp. 1–6.

[22] D. Cha and D. Kim, “DAM-GAN: Image inpainting using dynamic attention map based on fake texture detection,” in Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 4883–4887.

[23] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 4471–4480.

[24] K. Zhang, J. Fu, and D. Liu, “Inertia-guided flow completion and style fusion for video inpainting,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5982–5991.

[25] H. Veeraraghavan, N. Papanikolopoulos, and P. Schrater, “Deterministic sampling-based switching kalman filtering

for vehicle tracking," in Proceedings of the IEEE Intelligent Transportation Systems Conference, 2006, pp. 1340–1345.

[26] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Non-local social pooling for vehicle trajectory prediction." in Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 975–980.

[27] W. Xiao, J. Zou, H. Li, and K. Xu, "Smooth trajectory tracking using longitudinal distance constraint for a 4ws4wd unmanned ground vehicle," in Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019, pp. 2105–2110.

[28] S. Wang, D. Zhang, J. An, and J. Zhang, "Lane level turning trajectory tracking of intelligent vehicle based on drivers' manipulate habits," in Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), 2017, pp. 6873–6877.

[29] H. M. Mandalia and M. D. D. Salvucci, "Using support vector machines for lane-change detection," in Proceedings of the human factors and ergonomics society annual meeting, vol. 49, no. 22, 2005, pp. 1965–1969.

[30] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 1307–1312.

[31] S. Park, B. Kim, C. Kang, C. Chung, and J. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 1672–1678.

[32] Y. Cai, Z. Wang, H. Wang, L. Chen, Y. Li, M. Sotelo, and Z. Li, "Environment-attention network for vehicle trajectory prediction," IEEE Transactions on Vehicular Technology, vol. 70, no. 11, pp. 11 216–11 227, 2021.

[33] M. Arjovsky, S. Chintala, Lxe, and O. Bottou, "Wasserstein generative adversarial networks," in Proceedings of the International Conference on Machine Learning, 2017, pp. 214–223.

[34] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in Proceedings of the Advances in Neural Information Processing Systems, vol. 30, 2017.

[35] L. Zhang, C. Bao, and K. Ma, "Self-distillation: Towards efficient and compact neural networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 8, pp. 4388–4403, 2021.

[36] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2118–2125.

[37] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2575–2582.